Einrichten des Astra Control Center

Astra Control Center

NetApp
June 06, 2024

Inhalt

Einrichten des Astra Control Center

Astra Control Center unterstützt und überwacht ONTAP und Astra Data Store als Storage-Backend. Nach der Installation von Astra Control Center, melden Sie sich in der UI an und ändern Sie Ihr Passwort, Sie möchten eine Lizenz einrichten, Cluster hinzufügen, Speicher verwalten und Buckets hinzufügen.

Aufgaben

- Fügen Sie eine Lizenz für Astra Control Center hinzu
- Cluster hinzufügen
- Fügen Sie ein Storage-Back-End hinzu
- Fügen Sie einen Bucket hinzu

Fügen Sie eine Lizenz für Astra Control Center hinzu

Sie können eine neue Lizenz über die UI oder hinzufügen "API" Um die Funktionalität des Astra Control Center voll zu nutzen. Ohne Lizenz ist Ihre Verwendung von Astra Control Center auf das Management von Benutzern und das Hinzufügen neuer Cluster beschränkt.

Weitere Informationen zur Berechnung von Lizenzen finden Sie unter "Lizenzierung".

> Informationen zum Aktualisieren einer vorhandenen Testversion oder Volllizenz finden Sie unter "Aktualisieren einer vorhandenen Lizenz".

Astra Control Center Lizenzen messen die CPU-Ressourcen mithilfe von Kubernetes CPU-Einheiten. Die Lizenz muss die CPU-Ressourcen berücksichtigen, die den Worker-Nodes aller verwalteten Kubernetes-Cluster zugewiesen sind. Bevor Sie eine Lizenz hinzufügen, müssen Sie die Lizenzdatei (NLF) vom beziehen "NetApp Support Website".

Sie können das Astra Control Center auch mit einer Evaluierungslizenz ausprobieren, mit der Sie das Astra Control Center 90 Tage ab dem Tag, an dem Sie die Lizenz herunterladen, nutzen können. Sie können sich durch die Anmeldung für eine kostenlose Testversion anmelden "Hier".

> Wenn Ihre Installation die Anzahl der lizenzierten CPU-Einheiten überschreitet, verhindert Astra Control Center die Verwaltung neuer Anwendungen. Bei Überschreitung der Kapazität wird eine Meldung angezeigt.

Was Sie benötigen

Wenn Sie Astra Control Center von heruntergeladen "NetApp Support Website", Sie haben auch die NetApp Lizenzdatei (NLF) heruntergeladen. Stellen Sie sicher, dass Sie Zugriff auf diese Lizenzdatei haben.

Schritte

1. Melden Sie sich in der UI des Astra Control Center an.
2. Wählen Sie **Konto** > **Lizenz**.
3. Wählen Sie **Lizenz Hinzufügen**.
4. Rufen Sie die Lizenzdatei (NLF) auf, die Sie heruntergeladen haben.
5. Wählen Sie **Lizenz Hinzufügen**.

Auf der Seite **Konto** > **Lizenz** werden Lizenzinformationen, Ablaufdatum, Lizenzseriennummer, Konto-ID und

verwendete CPU-Einheiten angezeigt.

Wenn Sie über eine Evaluierungslizenz verfügen, sollten Sie Ihre Konto-ID speichern, um Datenverlust im Falle eines Ausfalls des Astra Control Center zu vermeiden, wenn Sie ASUPs nicht senden.

Cluster hinzufügen

Zum Management von Applikationen fügen Sie einen Kubernetes-Cluster hinzu und managen ihn als Computing-Ressource. Um Ihre Kubernetes-Applikationen zu ermitteln, müssen Sie einen Cluster hinzufügen, in dem Astra Control Center ausgeführt werden kann. Bei Astra Data Store möchten Sie den Kubernetes App-Cluster hinzufügen, der Applikationen enthält, die von Astra Data Store bereitgestellte Volumes verwenden.

Wir empfehlen, dass Astra Control Center den Cluster, der zuerst bereitgestellt wird, verwaltet, bevor Sie zum Management weitere Cluster zum Astra Control Center hinzufügen. Das Management des anfänglichen Clusters ist erforderlich, um Kubemetrics-Daten und Cluster-zugeordnete Daten zur Metriken und Fehlerbehebung zu senden. Sie können die * Cluster hinzufügen* Funktion verwenden, um einen Cluster mit Astra Control Center zu verwalten.

Wenn Astra Control einen Cluster verwaltet, wird die Standard-Storage-Klasse des Clusters überwacht. Wenn Sie die Speicherklasse mit ändern `kubectl` Befehle, Astra Control setzt die Änderung zurück. Verwenden Sie eine der folgenden Methoden, um die Standard-Storage-Klasse in einem von Astra Control gemanagten Cluster zu ändern:

- Verwenden Sie die Astra Control API `PUT /managedClusters` endpunkt, und weisen Sie dem eine andere Standard-Speicherklasse zu `DefaultStorageClass` Parameter.
- Verwenden Sie die Web-UI von Astra Control, um eine andere Standard-Storage-Klasse zuzuweisen. Siehe Ändern der Standard-Storage-Klasse.

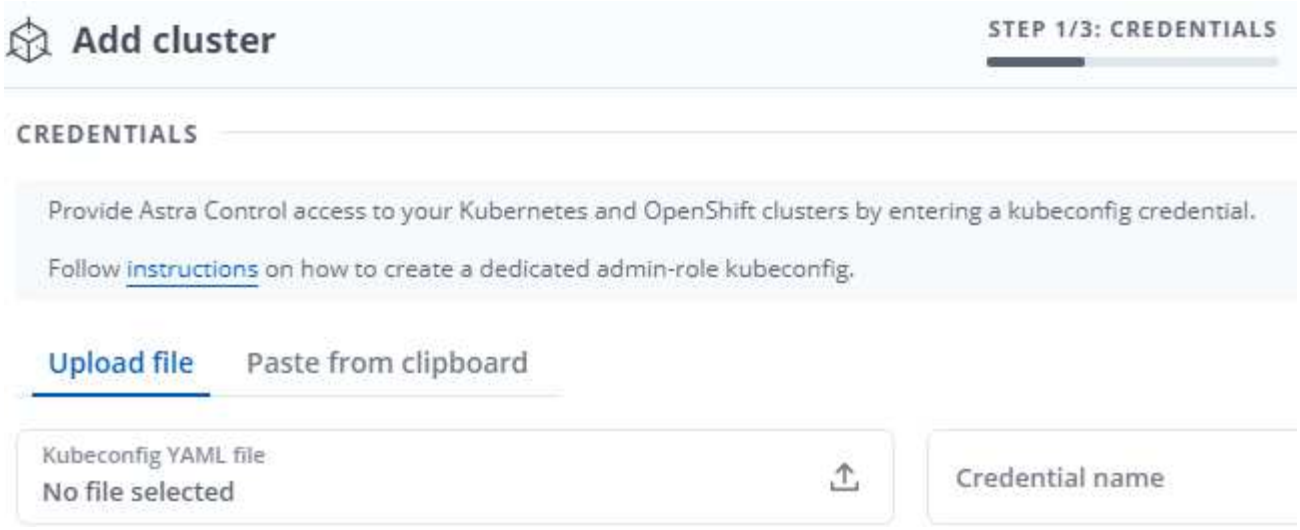
Was Sie benötigen

- Bevor Sie ein Cluster hinzufügen, überprüfen und führen Sie die erforderlichen Maßnahmen durch "Erforderliche Aufgaben".

Schritte

1. Wählen Sie in der Astra Control Center-Benutzeroberfläche auf dem Dashboard * im Bereich Cluster die Option **Add** aus.
2. Laden Sie im Fenster **Cluster hinzufügen** ein `kubeconfig.yaml` Datei oder fügen Sie den Inhalt eines ein `kubeconfig.yaml` Datei:

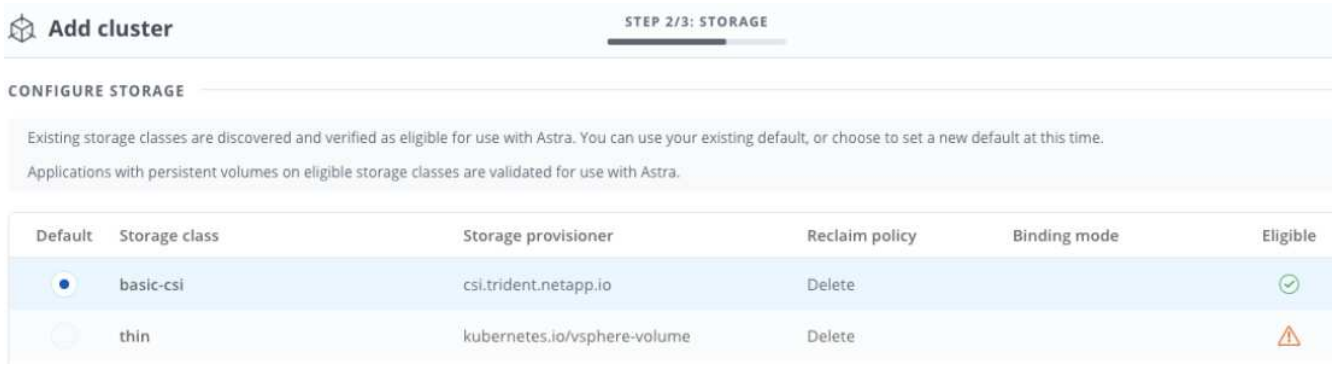
   Der `kubeconfig.yaml` Die Datei sollte **nur die Cluster-Anmeldedaten für einen Cluster** enthalten.



Wenn Sie Ihre eigenen erstellen `kubeconfig` Datei, Sie sollten nur ein **ein**-Kontext -Element darin definieren. Siehe "Kubernetes-Dokumentation" Weitere Informationen zum Erstellen `kubeconfig` Dateien:

3. Geben Sie einen Namen für die Anmeldeinformationen an. Standardmäßig wird der Name der Anmeldeinformationen automatisch als Name des Clusters ausgefüllt.
4. Wählen Sie * Storage konfigurieren* aus.
5. Wählen Sie die Storage-Klasse aus, die für diesen Kubernetes-Cluster verwendet werden soll, und wählen Sie **Review** aus.

Sie sollten sich für einen Trident-Storage-Kurs entscheiden, der von ONTAP Storage oder Astra Data Store unterstützt wird.



6. Überprüfen Sie die Informationen, und wenn alles gut aussieht, wählen Sie **Cluster hinzufügen**.

Ergebnis

Der Cluster wechselt in den **Entdeckungs**-Status und dann in **running**. Sie haben erfolgreich ein Kubernetes-Cluster hinzugefügt und verwalten es jetzt im Astra Control Center.

Nachdem Sie einen Cluster hinzugefügt haben, der im Astra Control Center verwaltet werden soll, kann es in einigen Minuten dauern, bis der Monitoring-Operator implementiert ist. Bis dahin wird das Benachrichtigungssymbol rot und ein Ereignis **Überwachung Agent-Status-Prüfung fehlgeschlagen** protokolliert. Sie können dies ignorieren, da das Problem gelöst wird, wenn Astra Control Center den richtigen Status erhält. Wenn sich das Problem in wenigen Minuten nicht beheben lässt, wechseln Sie zum Cluster und führen Sie aus `oc get pods -n netapp-monitoring` Als Ausgangspunkt. Um das Problem zu beheben, müssen Sie sich die Protokolle des Überwachungsperbers ansehen.

Fügen Sie ein Storage-Back-End hinzu

Sie können ein Storage-Backend hinzufügen, sodass Astra Control die Ressourcen managen kann. Sie können ein Storage-Back-End auf einem gemanagten Cluster implementieren oder ein vorhandenes Storage-Back-End verwenden.

Durch das Management von Storage-Clustern in Astra Control als Storage-Backend können Sie Verbindungen zwischen persistenten Volumes (PVS) und dem Storage-Backend sowie zusätzliche Storage-Kennzahlen abrufen.

Was Sie für bestehende Implementierungen von Astra Data Store benötigen

- Sie haben Ihren Kubernetes-App-Cluster und das zugrunde liegende Computing-Cluster hinzugefügt.

Nachdem Sie Ihren Kubernetes App-Cluster für Astra Data Store hinzugefügt haben und er durch Astra Control gemanagt wird, erscheint der Cluster wie `unmanaged` In der Liste der entdeckten Back-Ends. Als Nächstes müssen Sie das Computing-Cluster hinzufügen, das Astra Data Store enthält und das Kubernetes App-Cluster untermauert. Dies können Sie über **Backends** in der UI tun. Wählen Sie das Menü Aktionen für den Cluster aus `Manage`, und "Fügen Sie den Cluster hinzu". Nach dem Status des Clusters von `unmanaged` Änderungen am Namen des Kubernetes-Clusters können Sie mit dem Hinzufügen eines Backend fortfahren.

Was Sie für die neuen Astra Data Store Implementierungen benötigen

- Das ist schon "Die Version des Installationspakets, das Sie bereitstellen möchten, hochgeladen haben" Zu einem Ort, der für Astra Control zugänglich ist.
- Sie haben den Kubernetes-Cluster hinzugefügt, den Sie für die Implementierung verwenden möchten.
- Sie haben die hochgeladen Astra Data Store-Lizenz Für Ihre Implementierung an einen Standort, auf den Astra Control zugreifen kann.

Optionen

- Implementieren von Storage-Ressourcen
- Verwenden Sie ein vorhandenes Storage-Back-End

Implementieren von Storage-Ressourcen

Sie können einen neuen Astra Data Store implementieren und das zugehörige Storage-Backend verwalten.

Schritte

1. Navigieren Sie im Dashboard oder im Menü „Backend“:
    - Aus **Dashboard**: Wählen Sie in der Ressourcenübersicht einen Link aus dem Fensterbereich

Speicherrückseite aus und wählen Sie im Bereich Back Ends **Add** aus.

- Von **Backends**:
  i. Wählen Sie im linken Navigationsbereich **Backend** aus.
  ii. Wählen Sie **Hinzufügen**.

2. Wählen Sie auf der Registerkarte **Bereitstellen** die Option * Astra Data Store* Deployment aus.
3. Wählen Sie das zu implementierende Astra Data Store-Paket aus:
   a. Geben Sie einen Namen für die Astra Data Store-Anwendung ein.
   b. Wählen Sie die Version des Astra Data Stores, die Sie implementieren möchten.

> Wenn Sie die Version, die Sie bereitstellen möchten, noch nicht hochgeladen haben, können Sie die Option **Paket hinzufügen** verwenden oder den Assistenten beenden und verwenden "Paketmanagement" Um das Installationspaket hochzuladen.

4. Wählen Sie eine Astra Data Store-Lizenz aus, die Sie bereits hochgeladen haben, oder laden Sie die **Lizenz hinzufügen**-Option ein, die Sie mit der Anwendung verwenden können.

> Astra Data Store-Lizenzen mit vollständigen Berechtigungen sind mit Ihrem Kubernetes-Cluster verknüpft. Die zugehörigen Cluster sollten automatisch angezeigt werden. Wenn kein verwalteter Cluster vorhanden ist, können Sie die Option **Cluster hinzufügen** zur Astra Control-Verwaltung hinzufügen wählen. Für Astra Data Store-Lizenzen können Sie diese Verknüpfung auf der nächsten Seite des Assistenten definieren, wenn keine Verbindung zwischen Lizenz und Cluster hergestellt wurde.

5. Wenn Sie dem Astra Control Management noch kein Kubernetes-Cluster hinzugefügt haben, müssen Sie dies auf der Seite * Kubernetes Cluster* tun. Wählen Sie einen vorhandenen Cluster aus der Liste aus, oder wählen Sie **Hinzufügen des zugrunde liegenden Clusters** aus, um ein Cluster zum Astra Control Management hinzuzufügen.
6. Wählen Sie die Größe der Implementierungsvorlage für den Kubernetes Cluster aus, die Ressourcen für Astra Data Store bereitstellen soll.

> Wählen Sie bei der Auswahl einer Vorlage größere Nodes mit mehr Arbeitsspeicher und Kernen für größere Workloads oder eine größere Anzahl an Nodes für kleinere Workloads aus. Wählen Sie eine Vorlage basierend auf den von Ihrer Lizenz zulässt aus. Bei jeder Vorlagenoption werden die Anzahl der qualifizierten Nodes angegeben, die das Vorlagenmuster für Arbeitsspeicher und Kerne sowie die Kapazität für jeden Node erfüllen.

7. Konfigurieren der Nodes:
   a. Fügen Sie eine Node-Bezeichnung hinzu, um den Pool der Worker-Nodes zu identifizieren, die diesen Astra Data Store-Cluster unterstützen.

> Das Label muss jedem einzelnen Node im Cluster hinzugefügt werden, der vor Beginn der Implementierung oder der Implementierung von Astra Data Store genutzt wird.

   b. Konfigurieren Sie die Kapazität (gib) pro Node manuell, oder wählen Sie die maximal zulässige Node-Kapazität aus.
   c. Konfigurieren Sie eine Höchstzahl der im Cluster zulässigen Nodes oder zulassen die maximale Anzahl der Nodes im Cluster.

8. (Nur Astra Data Store Volllizenzen) Geben Sie den Schlüssel des Etiketts ein, das Sie für Protection Domains verwenden möchten.

> Erstellen Sie für jeden Node mindestens drei eindeutige Beschriftungen für den Schlüssel. Beispiel: Wenn Ihr Schlüssel lautet `astra.datastore.protection.domain`, Sie können die folgenden Etiketten erstellen: `astra.datastore.protection.domain=domain1`,`astra.datastore.protection.domain=domain2`, und `astra.datastore.protection.domain=domain3`.

9. Konfigurieren des Managementnetzwerks:
   a. Geben Sie eine Management-IP-Adresse für die interne Verwaltung von Astra Data Store ein, die sich im gleichen Subnetz wie die IP-Adressen der Worker-Nodes befindet.
   b. Sie können dieselbe NIC sowohl für Management- als auch für Datennetzwerke verwenden oder sie separat konfigurieren.
   c. Geben Sie den Daten-Netzwerk-IP-Adressenpool, die Subnetzmaske und das Gateway für den Storage-Zugriff ein.
10. Überprüfen Sie die Konfiguration und wählen Sie **Bereitstellen**, um mit der Installation zu beginnen.

Ergebnis

Nach erfolgreicher Installation erscheint das Backend in `available` Geben Sie in der Back-Ends-Liste zusammen mit aktiven Performance-Informationen an.

> Möglicherweise müssen Sie die Seite aktualisieren, damit das Backend angezeigt wird.

Verwenden Sie ein vorhandenes Storage-Back-End

Sie können ein entdecktes ONTAP oder Astra Data Store Storage Back-End in das Astra Control Center Management integrieren.

Schritte

1. Navigieren Sie im Dashboard oder im Menü „Backend“:
   - Aus **Dashboard**: Wählen Sie in der Ressourcenübersicht einen Link aus dem Fensterbereich Speicherrückseite aus und wählen Sie im Bereich Back Ends **Add** aus.
   - Von **Backends**:
     i. Wählen Sie im linken Navigationsbereich **Backend** aus.
     ii. Wählen Sie **Verwalten** auf einem ermittelten Backend aus dem verwalteten Cluster oder wählen Sie **Hinzufügen**, um ein zusätzliches vorhandenes Backend zu verwalten.
2. Wählen Sie die Registerkarte **vorhandene** verwenden.
3. Je nach Backend-Typ:
   - **Astra Data Store**:
     i. Wählen Sie **Astra Data Store**.
     ii. Wählen Sie das verwaltete Compute-Cluster aus und wählen Sie **Next** aus.
     iii. Bestätigen Sie die Back-End-Details und wählen Sie **Add Storage Backend**.
   - **ONTAP**:
     i. Wählen Sie **ONTAP**.

ii. Geben Sie die Anmeldedaten für den ONTAP-Administrator ein, und wählen Sie **Überprüfen**.

iii. Bestätigen Sie die Back-End-Details und wählen Sie **Add Storage Backend**.

Ergebnis

Das Backend wird in angezeigt `available` Status in der Liste mit Zusammenfassungsinformationen.

Möglicherweise müssen Sie die Seite aktualisieren, damit das Backend angezeigt wird.

Fügen Sie einen Bucket hinzu

Das Hinzufügen von Objektspeicher-Bucket-Providern ist wichtig, wenn Sie Ihre Applikationen und Ihren persistenten Storage sichern möchten oder Applikationen über Cluster hinweg klonen möchten. Astra Control speichert diese Backups oder Klone in den von Ihnen definierten Objektspeicher-Buckets.

Wenn Sie einen Bucket hinzufügen, markiert Astra Control einen Bucket als Standard-Bucket-Indikator. Der erste von Ihnen erstellte Bucket wird der Standard-Bucket.

Sie brauchen keinen Eimer, wenn Sie Ihre Anwendungskonfiguration und Ihren persistenten Storage im selben Cluster klonen.

Verwenden Sie einen der folgenden Bucket-Typen:

- NetApp ONTAP S3
- NetApp StorageGRID S3
- Allgemein S3

Obwohl Astra Control Center Amazon S3 als Generic S3 Bucket-Provider unterstützt, unterstützt Astra Control Center möglicherweise nicht alle Objektspeicher-Anbieter, die die S3-Unterstützung von Amazon beanspruchen.

Anweisungen zum Hinzufügen von Buckets mithilfe der Astra Control API finden Sie unter "Astra Automation und API-Informationen".

Schritte

1. Wählen Sie im linken Navigationsbereich **Buckets** aus.
   a. Wählen Sie **Hinzufügen**.
   b. Wählen Sie den Bucket-Typ aus.

      Wenn Sie einen Bucket hinzufügen, wählen Sie den richtigen Bucket-Provider aus und geben die richtigen Anmeldedaten für diesen Provider an. Beispielsweise akzeptiert die UI NetApp ONTAP S3 als Typ und akzeptiert StorageGRID-Anmeldedaten. Dies führt jedoch dazu, dass alle künftigen Applikations-Backups und -Wiederherstellungen, die diesen Bucket verwenden, fehlschlagen.

   c. Erstellen Sie einen neuen Bucket-Namen oder geben Sie einen vorhandenen Bucket-Namen und eine optionale Beschreibung ein.

Der Bucket-Name und die Beschreibung erscheinen als Backup-Speicherort, den Sie später wählen können, wenn Sie ein Backup erstellen. Der Name wird auch während der Konfiguration der Schutzrichtlinien angezeigt.

d. Geben Sie den Namen oder die IP-Adresse des S3-Endpunkts ein.

e. Wenn dieser Bucket der Standard-Bucket für alle Backups sein soll, prüfen Sie den `Make this bucket the default bucket for this private cloud` Option.

Diese Option wird nicht für den ersten von Ihnen erstellten Bucket angezeigt.

f. Mit Hinzufügen fortfahren Anmeldeinformationen.

Fügen Sie S3-Zugriffsdaten hinzu

Fügen Sie Ihre Zugangsdaten für S3-Zugriff jederzeit hinzu.

Schritte

1. Wählen Sie im Dialogfeld Buckets entweder die Registerkarte **Hinzufügen** oder **vorhandene verwenden** aus.
   a. Geben Sie einen Namen für die Anmeldedaten ein, der sie von anderen Anmeldeinformationen in Astra Control unterscheidet.
   b. Geben Sie die Zugriffs-ID und den geheimen Schlüssel ein, indem Sie den Inhalt aus der Zwischenablage einfügen.

Ändern der Standard-Storage-Klasse

Sie können die Standard-Storage-Klasse für ein Cluster ändern.

Schritte

1. Wählen Sie in der Web-UI des Astra Control Center die Option **Cluster** aus.
2. Wählen Sie auf der Seite **Cluster** den Cluster aus, den Sie ändern möchten.
3. Wählen Sie die Registerkarte **Storage** aus.
4. Wählen Sie die Kategorie **Speicherklassen** aus.
5. Wählen Sie das Menü **Aktionen** für die Speicherklasse aus, die Sie als Standard festlegen möchten.
6. Wählen Sie **als Standard**.

Was kommt als Nächstes?

Nachdem Sie sich angemeldet haben und Cluster zum Astra Control Center hinzugefügt haben, können Sie die Anwendungsdatenmanagement-Funktionen von Astra Control Center nutzen.

- "Benutzer managen"
- "Starten Sie das Anwendungsmanagement"
- "Schützen von Applikationen"
- "Applikationen klonen"

- "Benachrichtigungen verwalten"
- "Verbinden Sie sich mit Cloud Insights"
- "Fügen Sie ein benutzerdefiniertes TLS-Zertifikat hinzu"

Weitere Informationen

- "Verwenden Sie die Astra Control API"
- "Bekannte Probleme"

Voraussetzungen für das Hinzufügen eines Clusters

Sie sollten sicherstellen, dass die Voraussetzungen erfüllt sind, bevor Sie ein Cluster hinzufügen. Außerdem sollten Sie die Eignungskontrollen durchführen, um sicherzustellen, dass Ihr Cluster zum Astra Control Center hinzugefügt werden kann.

Was benötigen Sie vor dem Hinzufügen eines Clusters

- Einer der folgenden Cluster-Typen:
  - Cluster mit OpenShift 4.6.8, 4.7, 4.8 oder 4.9
  - Cluster mit Rancher 2.5.8, 2.5.9 oder 2.6 mit RKE1
  - Cluster laufen mit Kubernetes 1.20 bis 1.23
  - Cluster mit VMware Tanzu Kubernetes Grid 1.4
  - Cluster mit VMware Tanzu Kubernetes Grid Integrated Edition 1.12.2

    Stellen Sie sicher, dass auf Ihren Clustern ein oder mehrere Worker-Nodes mit mindestens 1 GB RAM für laufende Telemetrieservices verfügbar sind.

> Wenn Sie planen, als verwaltete Computing-Ressource einen zweiten OpenShift 4.6, 4.7 oder 4.8 hinzuzufügen, sollten Sie sicherstellen, dass die Astra Trident Volume Snapshot-Funktion aktiviert ist. Sehen Sie den offiziellen Astra Trident an "Anweisungen" Um Volume Snapshots mit Astra Trident zu aktivieren und zu testen.

- Astra Trident StorageClasses ist mit einem konfiguriert "Unterstütztes Storage-Backend" (Erforderlich für jeden Cluster-Typ)
- Der Superuser und die Benutzer-ID sind auf dem ONTAP-System eingerichtet, um Apps mit Astra Control Center zu sichern und wiederherzustellen. Führen Sie den folgenden Befehl in der ONTAP-Befehlszeile aus:
  `export-policy rule modify -vserver <storage virtual machine name> -policyname <policy name> -ruleindex 1 -superuser sysm --anon 65534`
- Astra Trident `volumesnapshotclass` Objekt, das von einem Administrator definiert wurde. Astra Trident ist der Anfang "Anweisungen" Um Volume Snapshots mit Astra Trident zu aktivieren und zu testen.
- Stellen Sie sicher, dass nur eine einzelne Standard-Storage-Klasse für Ihr Kubernetes-Cluster definiert ist.

Führen Sie Eignungsprüfungen durch

Führen Sie die folgenden Eignungsprüfungen durch, um sicherzustellen, dass Ihr Cluster zum Astra Control

Center hinzugefügt werden kann.

Schritte

1. Überprüfen Sie die Trident Version.

```
kubectl get tridentversions -n trident
```

Wenn Trident vorhanden ist, wird eine Ausgabe ähnlich der folgenden ausgegeben:

```
NAME      VERSION
trident   21.04.0
```

Wenn Trident nicht vorhanden ist, wird eine Ausgabe wie die folgende angezeigt:

```
error: the server doesn't have a resource type "tridentversions"
```

> Wenn Trident nicht installiert ist oder die installierte Version nicht die neueste ist, müssen Sie die neueste Version von Trident installieren, bevor Sie fortfahren. Siehe "Trident Dokumentation" Weitere Anweisungen.

2. Prüfen Sie, ob die Storage-Klassen die unterstützten Trident Treiber verwenden. Der bereitstellungsname sollte lauten `csi.trident.netapp.io`. Das folgende Beispiel zeigt:

```
kubectl get sc
NAME                     PROVISIONER                    RECLAIMPOLICY
VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
ontap-gold (default)   csi.trident.netapp.io          Delete
Immediate           true                   5d23h
thin                   kubernetes.io/vsphere-volume   Delete
Immediate           false                  6d
```

Erstellen Sie ein „admin-Role“-kubeconfig

Stellen Sie sicher, dass Sie die folgenden Schritte auf Ihrem Gerät ausführen:

- `kubectl` v1.19 oder höher installiert
- Ein aktiver kubeconfig mit Clusteradministratorrechten für den aktiven Kontext

Schritte

1. Erstellen Sie ein Service-Konto wie folgt:

   a. Erstellen Sie eine Dienstkontendatei mit dem Namen `astracontrol-service-account.yaml`.

      Passen Sie Namen und Namespace nach Bedarf an. Wenn hier Änderungen vorgenommen werden,

sollten Sie die gleichen Änderungen in den folgenden Schritten anwenden.

```
<strong>astracontrol-service-account.yaml</strong>
```

+

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: astracontrol-service-account
  namespace: default
```

a. Wenden Sie das Servicekonto an:

```
kubectl apply -f astracontrol-service-account.yaml
```

2. (Optional) Wenn Ihr Cluster eine restriktive Pod-Sicherheitsrichtlinie verwendet, die die Erstellung privilegierter Pods nicht zulässt oder Vorgänge innerhalb der Pod-Container als Root-Benutzer ausgeführt werden können, erstellen Sie eine benutzerdefinierte Pod-Sicherheitsrichtlinie für den Cluster, durch die Astra Control Pods erstellen und managen kann. Anweisungen hierzu finden Sie unter "Erstellen einer benutzerdefinierten POD-Sicherheitsrichtlinie".
3. Gewähren Sie Cluster-Admin-Berechtigungen wie folgt:

a. Erstellen Sie ein `ClusterRoleBinding` Datei aufgerufen `astracontrol-clusterrolebinding.yaml`.

Passen Sie bei Bedarf alle beim Erstellen des Dienstkontos geänderten Namen und Namespaces an.

```
<strong>astracontrol-clusterrolebinding.yaml</strong>
```

+

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: astracontrol-admin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: astracontrol-service-account
  namespace: default
```

a. Wenden Sie die Bindung der Cluster-Rolle an:

```
kubectl apply -f astracontrol-clusterrolebinding.yaml
```

4. Listen Sie die Geheimnisse des Dienstkontos auf, ersetzen Sie `<context>` Mit dem richtigen Kontext für Ihre Installation:

```
kubectl get serviceaccount astracontrol-service-account --context
<context> --namespace default -o json
```

Das Ende der Ausgabe sollte wie folgt aussehen:

```
"secrets": [
{ "name": "astracontrol-service-account-dockercfg-vhz87"},
{ "name": "astracontrol-service-account-token-r59kr"}
]
```

Die Indizes für jedes Element im `secrets` Array beginnt mit 0. Im obigen Beispiel der Index für `astracontrol-service-account-dockercfg-vhz87` Wäre 0 und der Index für `astracontrol-service-account-token-r59kr` Sind es 1. Notieren Sie in Ihrer Ausgabe den Index für den Namen des Dienstkontos, der das Wort „Token“ darin enthält.

5. Erzeugen Sie den kubeconfig wie folgt:

a. Erstellen Sie ein `create-kubeconfig.sh` Datei: Austausch `TOKEN_INDEX` Am Anfang des folgenden Skripts mit dem korrekten Wert.

```
<strong>create-kubeconfig.sh</strong>
```

```
```

```
# Update these to match your environment.
# Replace TOKEN_INDEX with the correct value
# from the output in the previous step. If you
# didn't change anything else above, don't change
# anything else here.

SERVICE_ACCOUNT_NAME=astracontrol-service-account
NAMESPACE=default
NEW_CONTEXT=astracontrol
KUBECONFIG_FILE='kubeconfig-sa'

CONTEXT=$(kubectl config current-context)

SECRET_NAME=$(kubectl get serviceaccount ${SERVICE_ACCOUNT_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.secrets[TOKEN_INDEX].name}')
TOKEN_DATA=$(kubectl get secret ${SECRET_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.data.token}')

TOKEN=$(echo ${TOKEN_DATA} | base64 -d)

# Create dedicated kubeconfig
# Create a full copy
kubectl config view --raw > ${KUBECONFIG_FILE}.full.tmp

# Switch working context to correct context
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp config use-context
${CONTEXT}

# Minify
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp \
  config view --flatten --minify > ${KUBECONFIG_FILE}.tmp

# Rename context
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  rename-context ${CONTEXT} ${NEW_CONTEXT}

# Create token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-credentials ${CONTEXT}-${NAMESPACE}-token-user \
  --token ${TOKEN}

# Set context to use token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
```

```
  set-context ${NEW_CONTEXT} --user ${CONTEXT}-${NAMESPACE}-token
-user

# Set context to correct namespace
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --namespace ${NAMESPACE}

# Flatten/minify kubeconfig
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  view --flatten --minify > ${KUBECONFIG_FILE}

# Remove tmp
rm ${KUBECONFIG_FILE}.full.tmp
rm ${KUBECONFIG_FILE}.tmp
```

b. Geben Sie die Befehle an, um sie auf Ihren Kubernetes-Cluster anzuwenden.

```
source create-kubeconfig.sh
```

6. (**Optional**) Umbenennen Sie die kubeconfig in einen aussagekräftigen Namen für Ihren Cluster. Schützen Sie die Cluster-Anmeldedaten.

```
chmod 700 create-kubeconfig.sh
mv kubeconfig-sa.txt YOUR_CLUSTER_NAME_kubeconfig
```

Was kommt als Nächstes?

Jetzt, wo du überprüft hast, dass die Voraussetzungen erfüllt sind, bist du bereit "Fügen Sie einen Cluster hinzu".

Weitere Informationen

- "Trident Dokumentation"
- "Verwenden Sie die Astra Control API"

Fügen Sie ein benutzerdefiniertes TLS-Zertifikat hinzu

Sie können das vorhandene selbst signierte TLS-Zertifikat entfernen und durch ein TLS-Zertifikat ersetzen, das von einer Zertifizierungsstelle (CA) signiert ist.

Was Sie benötigen

- Kubernetes-Cluster mit installiertem Astra Control Center
- Administratorzugriff auf eine Command Shell auf dem zu ausgeführten Cluster `kubectl` Befehle
- Private Schlüssel- und Zertifikatdateien aus der CA

Entfernen Sie das selbstsignierte Zertifikat

Entfernen Sie das vorhandene selbstsignierte TLS-Zertifikat.

1. Melden Sie sich mit SSH beim Kubernetes Cluster an, der als administrativer Benutzer Astra Control Center hostet.
2. Suchen Sie das mit dem aktuellen Zertifikat verknüpfte TLS-Geheimnis mit dem folgenden Befehl, Ersetzen `<ACC-deployment-namespace>` Mit dem Astra Control Center Deployment Namespace:

```
kubectl get certificate -n <ACC-deployment-namespace>
```

3. Löschen Sie den derzeit installierten Schlüssel und das Zertifikat mit den folgenden Befehlen:

```
kubectl delete cert cert-manager-certificates -n <ACC-deployment-
namespace>
kubectl delete secret secure-testing-cert -n <ACC-deployment-namespace>
```

Fügen Sie ein neues Zertifikat hinzu

Fügen Sie ein neues TLS-Zertifikat hinzu, das von einer CA signiert wird.

1. Verwenden Sie den folgenden Befehl, um das neue TLS-Geheimnis mit dem privaten Schlüssel und den Zertifikatdateien aus der CA zu erstellen und die Argumente in Klammern <> durch die entsprechenden Informationen zu ersetzen:

```
kubectl create secret tls <secret-name> --key <private-key-filename>
--cert <certificate-filename> -n <ACC-deployment-namespace>
```

2. Verwenden Sie den folgenden Befehl und das folgende Beispiel, um die Cluster-Datei CRD (Custom Resource Definition) zu bearbeiten und die zu ändern `spec.selfSigned` Mehrwert für `spec.ca.secretName` So verweisen Sie auf das zuvor erstellte TLS-Geheimnis:

```
kubectl edit clusterissuers.cert-manager.io/cert-manager-certificates -n
<ACC-deployment-namespace>
....

#spec:
#  selfSigned: {}

spec:
  ca:
    secretName: <secret-name>
```

3. Überprüfen Sie mit den folgenden Befehlen und der Beispiel-Ausgabe, ob die Änderungen korrekt sind und

das Cluster bereit ist, Zertifikate zu validieren, und ersetzen Sie sie `<ACC-deployment-namespace>` Mit dem Astra Control Center Deployment Namespace:

```
kubectl describe clusterissuers.cert-manager.io/cert-manager-
certificates -n <ACC-deployment-namespace>
....

Status:
  Conditions:
    Last Transition Time:  2021-07-01T23:50:27Z
    Message:               Signing CA verified
    Reason:                KeyPairVerified
    Status:                True
    Type:                  Ready
Events:                    <none>
```

4. Erstellen Sie die `certificate.yaml` Datei anhand des folgenden Beispiels, Ersetzen der Platzhalterwerte in Klammern <> durch entsprechende Informationen:

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: <certificate-name>
  namespace: <ACC-deployment-namespace>
spec:
  secretName: <certificate-secret-name>
  duration: 2160h # 90d
  renewBefore: 360h # 15d
  dnsNames:
  - <astra.dnsname.example.com> #Replace with the correct Astra Control
Center DNS address
  issuerRef:
    kind: ClusterIssuer
    name: cert-manager-certificates
```

5. Erstellen Sie das Zertifikat mit dem folgenden Befehl:

```
kubectl apply -f certificate.yaml
```

6. Überprüfen Sie mithilfe der folgenden Befehl- und Beispielausgabe, ob das Zertifikat korrekt erstellt wurde und mit den während der Erstellung angegebenen Argumenten (z. B. Name, Dauer, Verlängerungsfrist und DNS-Namen).

```
kubectl describe certificate -n <ACC-deployment-namespace>
....

Spec:
  Dns Names:
    astra.example.com
  Duration:  125h0m0s
  Issuer Ref:
    Kind:        ClusterIssuer
    Name:        cert-manager-certificates
  Renew Before:  61h0m0s
  Secret Name:   <certificate-secret-name>
Status:
  Conditions:
    Last Transition Time:  2021-07-02T00:45:41Z
    Message:               Certificate is up to date and has not expired
    Reason:                Ready
    Status:                True
    Type:                  Ready
  Not After:               2021-07-07T05:45:41Z
  Not Before:              2021-07-02T00:45:41Z
  Renewal Time:            2021-07-04T16:45:41Z
  Revision:                1
Events:                    <none>
```

7. Bearbeiten Sie die Option Ingress CRD TLS, um mit dem folgenden Befehl und Beispiel auf Ihr neues Zertifikatgeheimnis zu verweisen und die Platzhalterwerte in Klammern <> durch entsprechende Informationen zu ersetzen:

```
kubectl edit ingressroutes.traefik.containo.us -n <ACC-deployment-
namespace>
....

# tls:
#    options:
#      name: default
#    secretName: secure-testing-cert
#    store:
#      name: default

 tls:
    options:
      name: default
    secretName: <certificate-secret-name>
    store:
      name: default
```

8. Navigieren Sie mithilfe eines Webbrowsers zur IP-Adresse der Implementierung von Astra Control Center.
9. Vergewissern Sie sich, dass die Zertifikatdetails mit den Details des installierten Zertifikats übereinstimmen.
10. Exportieren Sie das Zertifikat und importieren Sie das Ergebnis in den Zertifikatmanager in Ihrem Webbrowser.

Erstellen einer benutzerdefinierten POD-Sicherheitsrichtlinie

Astra Control muss Kubernetes Pods auf den gemanagten Clustern erstellen und managen. Wenn Ihr Cluster eine restriktive Pod-Sicherheitsrichtlinie verwendet, die die Erstellung privilegierter Pods nicht zulässt oder Vorgänge innerhalb der Pod-Container nicht als Root-Benutzer ausgeführt werden können, müssen Sie eine weniger restriktive POD-Sicherheitsrichtlinie erstellen, damit Astra Control diese Pods erstellen und verwalten kann.

Schritte

1. Erstellen Sie eine Pod-Sicherheitsrichtlinie für den Cluster, die weniger restriktiv ist als der Standard, und speichern Sie sie in einer Datei. Beispiel:

```
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: astracontrol
  annotations:
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
spec:
  privileged: true
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - '*'
  volumes:
  - '*'
  hostNetwork: true
  hostPorts:
  - min: 0
    max: 65535
  hostIPC: true
  hostPID: true
  runAsUser:
    rule: 'RunAsAny'
  seLinux:
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'RunAsAny'
  fsGroup:
    rule: 'RunAsAny'
```

2. Erstellen Sie eine neue Rolle für die POD-Sicherheitsrichtlinie.

```
kubectl-admin create role psp:astracontrol \
    --verb=use \
    --resource=podsecuritypolicy \
    --resource-name=astracontrol
```

3. Binden Sie die neue Rolle an das Dienstkonto.

```
kubectl-admin create rolebinding default:psp:astracontrol \
    --role=psp:astracontrol \
    --serviceaccount=astracontrol-service-account:default
```